

## RESEARCH

## Open Access

# CP decomposition approach to blind separation for DS-CDMA system using a new performance index

Awatif Rouijel<sup>1\*</sup>, Khalid Minaoui<sup>1</sup>, Pierre Comon<sup>2</sup> and Driss Aboutajdine<sup>1</sup>**Abstract**

In this paper, we present a canonical polyadic (CP) tensor decomposition isolating the scaling matrix. This has two major implications: (i) the problem conditioning shows up explicitly and could be controlled through a constraint on the so-called coherences and (ii) a performance criterion concerning the factor matrices can be exactly calculated and is more realistic than performance metrics used in the literature. Two new algorithms optimizing the CP decomposition based on gradient descent are proposed. This decomposition is illustrated by an application to direct-sequence code division multiplexing access (DS-CDMA) systems; computer simulations are provided and demonstrate the good behavior of these algorithms, compared to others in the literature.

**Keywords:** CP decomposition; Tensor; DS-CDMA; Blind separation; Optimization

## 1 Introduction

Blind source separation consists in estimating unknown signals observed from their mixture without knowing any information about them, except mild properties such as their independence. Early work on blind source separation was initiated by Jutten and Héroult [1,2] in the case of an instantaneous mixture. More recently, the use of multilinear algebra methods has attracted attention in several areas such as data mining, signal processing, and particularly in wireless communication systems, among others. Wireless communication data can sometimes be viewed as components of a high-order tensor (order strictly larger than 2). Solving the problem of source separation then means finding a decomposition of this tensor and determining its parameters. One of the most popular tensor decompositions is the canonical polyadic decomposition (CP), also known as parallel factor analysis (PARAFAC), which can be seen as an analog of the matrix singular value decomposition (SVD), since it decomposes the tensor into a sum of rank-one components [3-5]. This decomposition has been exploited and generalized in several works

for solving different signal processing problems [6,7] such as multi-array multi-sensor signal processing. The interest of the CP decomposition lies in its uniqueness under certain conditions. Typical algorithms for finding the CP components include alternating least squares (ALS) and descent algorithms [8,9], which do not isolate the scaling factor matrix. Herein, we propose two new optimization algorithms for CP tensor decomposition, which isolate the scaling matrix in the optimization process and offers the possibility to monitor the conditioning.

It is well known that loading matrices are identified up to column scaling. This indeterminacy is complicated to take into account, given that the product of all scaling matrices must be equal to the identity. For this reason, only approximate performance indices have been used so far by ignoring the last constraint. However, one can ask oneself whether it is possible to calculate the exact performance index: this is our second contribution. The present paper develops preliminary results appeared in [10] and includes performances obtained in the frame of direct-sequence code division multiplexing access (DS-CDMA) blind multiuser detection and estimation.

The rest of this paper is organized as follows. Section 2 presents notation, definitions, and properties of third-order tensors, and the exact CP decomposition problem is then stated. In Section 3, the low-rank approximation

\*Correspondence: [awatifrouijel@gmail.com](mailto:awatifrouijel@gmail.com)

<sup>1</sup>Mohammed V-Agdal University, LRIT Associated Unit to the CNRS-URAC N29, Avenue des nations unies, Agdal, BP. 554 Rabat-Chellah, Morocco  
Full list of author information is available at the end of the article

is formulated. Existence and uniqueness of this decomposition are also investigated. ALS and the two proposed algorithms are presented in Section 4. Section 5 is dedicated to the new performance criterion with a focus on an exact performance index calculation. In Section 6, we show the usefulness of our algorithms and the performances obtained. An application of these algorithms to CDMA transmission is then provided to illustrate the effectiveness of the latter.

## 2 Notation and preliminaries

### 2.1 Notations and definitions

Let us first introduce some essential notation. Scalars are denoted by lowercase letters, e.g.,  $a$ . Vectors are denoted by boldface lowercase letters, e.g.,  $\mathbf{a}$ ; matrices are denoted by boldface capital letters, e.g.,  $\mathbf{A}$ . Higher-order tensors are denoted by boldface Euler script letters, e.g.,  $\mathcal{T}$ . The  $p$ th column of a matrix  $\mathbf{A}$  is denoted  $\mathbf{a}_p$ , the  $(i, j)$  entry of a matrix  $\mathbf{A}$  is denoted by  $A_{ij}$ , and element  $(i, j, k)$  of a third-order tensor  $\mathcal{T}$  is denoted by  $T_{ijk}$ .  $\mathbf{1}$  will represent a vector containing ones, and  $\mathbf{I}$  the identity matrix.

**Definition 2.1.** The scalar product of two same-sized tensors  $\mathcal{X}, \mathcal{Y} \in \mathbb{C}^{I_1 \times I_2 \times \dots \times I_N}$  is defined as:

$$\langle \mathcal{X}, \mathcal{Y} \rangle = \sum_{i_1=1}^{I_1} \dots \sum_{i_N=1}^{I_N} X_{i_1, \dots, i_N} Y_{i_1, \dots, i_N}^* \quad (1)$$

where  $X_{i_1, \dots, i_N}$  is the  $(i_1, \dots, i_N)$  elements of the  $N^{\text{th}}$  order of tensor.

**Definition 2.2.** The outer (tensor) product between two tensor arrays  $\mathcal{X} \in \mathbb{C}^{I_1 \times I_2 \times \dots \times I_N}$  and  $\mathcal{Y} \in \mathbb{C}^{J_1 \times J_2 \times \dots \times J_M}$  of orders  $N$  and  $M$ , respectively, is denoted by  $\mathcal{Z} = \mathcal{X} \otimes \mathcal{Y} \in \mathbb{C}^{I_1 \times I_2 \times \dots \times I_N \times J_1 \times J_2 \times \dots \times J_M}$  and defined by:

$$Z_{i_1, \dots, i_N, j_1, \dots, j_M} = X_{i_1, \dots, i_N} Y_{j_1, \dots, j_M} \quad (2)$$

The symbol  $\otimes$  represents the tensor outer product. The outer product of two tensors is another tensor, the order of which is given by the sum of the orders of the two former tensors. Equation 2 is a generalization of the concept of outer product of two vectors, which yields itself a matrix (second-order tensor). The outer product of three vectors  $\mathbf{a} \in \mathbb{C}^I$  and  $\mathbf{b} \in \mathbb{C}^J$  and  $\mathbf{c} \in \mathbb{C}^K$  yields a third-order decomposable tensor  $\mathcal{Z} = \mathbf{a} \otimes \mathbf{b} \otimes \mathbf{c} \in \mathbb{C}^{I \times J \times K}$  where  $Z_{ijk} = a_i b_j c_k$ .

**Definition 2.3.** The rank of an arbitrary tensor  $\mathcal{T} \in \mathbb{C}^{I_1 \times I_2 \times \dots \times I_N}$ , denoted by  $R = \text{rank}(\mathcal{T})$ , is the minimal number of rank-1 tensors that yield  $\mathcal{T}$  in a linear combination.

Decomposable tensors have thus a rank equal to 1.

**Definition 2.4.** The Kruskal rank, or *krank*, of a matrix is the largest number  $j$  such that every set of  $j$  columns is linearly independent.

**Definition 2.5.** The Frobenius norm of a tensor  $\mathcal{T} \in \mathbb{C}^{I_1 \times I_2 \times \dots \times I_N}$  is defined as:

$$\|\mathcal{T}\|_F = \sqrt{\langle \mathcal{T}, \mathcal{T} \rangle} \quad (3)$$

**Definition 2.6.** The Khatri-Rao product between two matrices with the same number of columns,  $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_F] \in \mathbb{C}^{I \times F}$  and  $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_F] \in \mathbb{C}^{J \times F}$ , is defined as the column-wise Kronecker product:

$$\mathbf{A} \odot \mathbf{B} = [\mathbf{a}_1 \boxtimes \mathbf{b}_1, \dots, \mathbf{a}_F \boxtimes \mathbf{b}_F] \in \mathbb{C}^{IJ \times F} \quad (4)$$

where  $\boxtimes$  is the Kronecker product.

**Definition 2.7.** The coherence of a collection  $V = \{\mathbf{v}_1, \dots, \mathbf{v}_r\}$  of unit-norm vectors is defined as the maximal value of the modulus of cross scalar products. In other words, the coherence of the collection  $V$  is defined as:

$$\mu_V = \max_{p \neq q} |\mathbf{v}_p^H \mathbf{v}_q| \quad (5)$$

**Definition 2.8.** Let  $\mathbf{A} \in \mathbb{C}^{I \times J}$ , then  $\text{vec}\{\mathbf{A}\} \in \mathbb{C}^K$  denotes the column vector defined by:

$$(\text{vec}\{\mathbf{A}\})_{i+(j-1)I} = A_{ij} \quad (6)$$

where  $K = IJ$ .

### 2.2 Preliminaries

A tensor of order  $d$  is an object defined on a product between  $d$  linear spaces. Once the bases of these spaces are fixed, a  $d$ th order tensor can be represented by a  $d$ -way array (a hyper matrix) of coordinates [4]. The order of a tensor thus corresponds to the number of indices of the associated array. We are interested in decomposing a third-order tensor  $\mathcal{T}$  as:

$$\mathcal{T} = \sum_{r=1}^R \lambda_r \mathcal{D}(r) \quad (7)$$

where  $\mathcal{D}(r)$  are decomposable tensors, that is,  $\mathcal{D}(r) = \mathbf{a}_r \otimes \mathbf{b}_r \otimes \mathbf{c}_r$ . Denote by  $(\cdot)^T$  matrix transposition,  $\lambda_r$  real positive scaling factors,  $\boldsymbol{\lambda} = [\lambda_1, \dots, \lambda_R]^T$ , and  $R$  the tensor rank. Vectors  $\mathbf{a}_r$  (resp.  $\mathbf{b}_r$  and  $\mathbf{c}_r$ ) live in a linear space of dimension  $I$  (resp. dimension  $J$  and  $K$ ). Equivalently, decomposition (7) can be rewritten as a function of array entries:

$$T_{ijk} = \sum_{r=1}^R \lambda_r A_{ir} B_{jr} C_{kr}, \quad i \in \{1, \dots, I\}, j \in \{1, \dots, J\}, k \in \{1, \dots, K\} \quad (8)$$

where  $A_{ir}$  (resp.  $B_{jr}$  and  $C_{kr}$ ) denote the entries of vector  $\mathbf{a}_r$  (resp.  $\mathbf{b}_r$  and  $\mathbf{c}_r$ ). The above decomposition is called the canonical polyadic decomposition (CP) of tensor  $\mathcal{T}$ . The model (7) can be written in a compact form using the Khatri-Rao product, as:

$$\begin{aligned}\mathbf{T}_1^{I,KJ} &= \mathbf{A} \mathbf{A} (\mathbf{C} \odot \mathbf{B})^T, \\ \mathbf{T}_2^{J,KI} &= \mathbf{B} \mathbf{A} (\mathbf{C} \odot \mathbf{A})^T, \\ \mathbf{T}_3^{K,JI} &= \mathbf{C} \mathbf{A} (\mathbf{B} \odot \mathbf{A})^T.\end{aligned}$$

where  $\mathbf{T}_1^{I,KJ}$  (resp.  $\mathbf{T}_2^{J,KI}$  and  $\mathbf{T}_3^{K,JI}$ ) is the matrix of size  $I \times KJ$  (resp.  $J \times KI$  and  $K \times JI$ ) obtained by unfolding the array  $\mathcal{T}$  of size  $I \times J \times K$  in the first mode (resp. the second mode and the third mode), and  $\mathbf{A}$  is the  $R \times R$  diagonal matrix defined as  $\mathbf{A} = \text{Diag}\{\lambda_1, \dots, \lambda_R\}$ ; see [5] for further details on matrix unfoldings.

The explicit handwriting of decomposable tensors as given in (8) is subject to scale indeterminacies. In the tensor literature, optimization of the CP decomposition (8) has been carried out without isolating the scaling factor  $\mathbf{A}$ , which is generally included in one of the loading matrices, so that  $\mathbf{A} = \mathbf{I}$ . In [5], Kolda and Bader proposed to reduce the indeterminacies by normalizing the vectors and storing the norms in  $\mathbf{A}$ . Our first proposal is to pull the factors  $\lambda_r$  outside the product and calculate the optimal value of the scaling factor, which permits to monitor the conditioning of the problem. Scaling indeterminacies are then clearly reduced to unit modulus but are not completely fixed, hence the difficulty in estimating the identification error of loading matrices  $\mathbf{A}$ ,  $\mathbf{B}$  and  $\mathbf{C}$ . Our second proposal (Section 5) is to calculate the  $3R$  complex phases (reducing to signs in the real case).

### 3 Existence and uniqueness

The goal is to identify all parameters in the right hand side of (8), given the whole array  $\mathcal{T}$ . According to existing results [11-13], a third-order tensor (i.e., a three-way array) of rank  $R$  can be uniquely represented as sum of  $R$  rank-1 tensors, under certain conditions. Kruskal has demonstrated that the condition (9) is sufficient for uniqueness in CP decomposition [13], where  $k_A$  is the krank of  $\mathbf{A}$ . This means that matrices  $\mathbf{A}$ ,  $\mathbf{B}$ , and  $\mathbf{C}$  are unique up to permutation and (complex) scaling of their columns, under the Kruskal's condition:

$$k_A + k_B + k_C \geq 2R + 2 \quad (9)$$

For uniqueness, Harshman has shown that is sufficient to have  $\mathbf{A}$  and  $\mathbf{B}$  of full rank, and  $\mathbf{C}$  of krank  $\geq 2$  [3]. When  $1 < R \leq 2$ , the Kruskal and Harshman conditions are equivalent. For  $R > 2$ , Kruskal's condition may be satisfied even when Harshman's are not, and this condition is claimed to be only sufficient for  $R > 3$  [14]. However, observations are actually corrupted by noise, so that (8) does not hold exactly.

### 3.1 Low-rank approximation

In practice, the exact CP decomposition always exists but with a very large rank. Hence, it may not be physically meaningful, and additionally, it is generally not unique. It is consequently preferred to fit a multi-linear model of lower rank,  $R < \text{rank} \mathcal{T}$ , fixed in advance, so that we have to deal with an approximation problem. To estimate the parameters of the decomposition, we need to minimize the following cost function:

$$\Upsilon(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{A}) = \|\mathcal{T} - (\mathbf{A}, \mathbf{B}, \mathbf{C}) \cdot \mathbf{A}\|_F^2. \quad (10)$$

By convention  $(\mathbf{A}, \mathbf{B}, \mathbf{C}) \cdot \mathbf{A}$  denotes the tensor of rank  $R$  of coordinates  $\sum_{r=1}^R \lambda_r A_{ir} B_{jr} C_{kr}$ . Minimizing error (10) means finding the best rank- $R$  approximate of  $\mathcal{T}$  and its CP decomposition. The cost function (10) can also be written in three equivalent compact forms with respect to the three loading matrices:

$$\begin{aligned}\Upsilon(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{A}) &= \|\mathbf{T}_1^{I,KJ} - \mathbf{A} \mathbf{A} (\mathbf{C} \odot \mathbf{B})^T\|_F^2, \\ &= \|\mathbf{T}_2^{J,KI} - \mathbf{B} \mathbf{A} (\mathbf{C} \odot \mathbf{A})^T\|_F^2, \\ &= \|\mathbf{T}_3^{K,JI} - \mathbf{C} \mathbf{A} (\mathbf{B} \odot \mathbf{A})^T\|_F^2.\end{aligned}$$

### 3.2 Conditioning of the problem

Assuming that the matrices  $\mathbf{A}$ ,  $\mathbf{B}$ , and  $\mathbf{C}$  are given, we will calculate the optimal value of the scaling factor  $\mathbf{A}$ . This can be done by expanding the Frobenius norm in (10), which is a quadratic form in the entries of  $\mathbf{A}$  and canceling the gradient with respect to  $\lambda$  (see details in Appendix 1). Then, the following linear system is obtained:

$$\mathbf{G} \boldsymbol{\lambda} = \mathbf{f}, \quad (11)$$

where  $\mathbf{f}$  is  $R$ -dimensional vector defined by the contraction  $f_r = \sum_{ijk} T_{ijk} A_{ir} B_{jr} C_{kr}$ ,  $\mathbf{G}$  represents the  $R \times R$  Gram matrix defined by:

$$G_{pq} = (\mathbf{a}_p \boxtimes \mathbf{b}_p \boxtimes \mathbf{c}_p)^H (\mathbf{a}_q \boxtimes \mathbf{b}_q \boxtimes \mathbf{c}_q). \quad (12)$$

In view of matrix  $\mathbf{G}$ , we can see that coherences play a role in the conditioning of the problem. From Equations 11 to 12, and since diagonal entries of  $\mathbf{G}$  are equal to 1, it is indeed clear that imposing cross scalar products of the form  $\mathbf{a}_p^H \mathbf{a}_q$  to have a modulus strictly smaller than 1 will lead with greater chances to an acceptable conditioning. Also note that scalar products do not appear individually in (12) but through their products, since entries of  $\mathbf{G}$  can also be written as  $G_{pq} = \mathbf{a}_p^H \mathbf{a}_q \mathbf{b}_p^H \mathbf{b}_q \mathbf{c}_p^H \mathbf{c}_q$ . This statement has deeper implications, particularly in existence and uniqueness of the solution to Problem (10), as subsequently elaborated.

### 3.3 Existence

According to the results in [15,16], the infimum of (10) may not be reachable. In fact, the set of tensors of rank at most  $\xi$  is not closed if  $\xi > 1$ . Examples of the lack of closeness have been provided in the literature [15,16], which suffice to prove it. In other words, it may happen that for a given tensor, and for any rank- $r$  approximation of it, there always exists another better rank- $r$  approximation.

This is the reason why the authors proposed the constraint below, which ensures existence of a minimum. Define the three coherences  $\mu_A$ ,  $\mu_B$ , and  $\mu_C$  associated with the matrices **A**, **B**, and **C**, respectively. It has been indeed shown in [15,17] that under the constraint:

$$\mu_A \mu_B \mu_C \leq \frac{1}{R-1}, \quad (13)$$

the infimum of (10) is reached. It may be seen that this condition already gives a quantitative bound to the conditioning of (11) because coherences bound extra-diagonal entries of matrix **G**, which has ones on its diagonal.

### 3.4 Uniqueness

The uniqueness of the tensor decomposition can be ensured by using a sufficient condition based on Kruskal's theorem (9), previously mentioned.

According to the lemma reported in [17,18], an inequality holds between Kruskal and coherence, namely  $k_A \geq \frac{1}{\mu_A}$ , as long as  $k_A$  is strictly smaller than the column rank of **A**. Including this inequality in Equation 9 leads to the following sufficient uniqueness condition:

$$\mu_A^{-1} + \mu_B^{-1} + \mu_C^{-1} \geq 2(R+1). \quad (14)$$

## 4 Optimization for CP decomposition

In Section 3, we presented CP for three-way tensors. Various optimization algorithms exist to compute CP decomposition without constraint, as ALS or descent algorithms [7,8,19]. We subsequently present optimization algorithms to compute the CP decomposition (10), under the constraints of unit norm columns of loading matrices.

### 4.1 Alternating least squares algorithm

The ALS algorithm was proposed for CP computation by Carroll and Chang in [20] and Harshman in [3] and still stays the workhorse algorithm today, mainly owing to its ease of implementation [21]. ALS is hence the classical solution to minimize the cost function (10), despite its lack of convergence proof. This iterative algorithm alternates among the estimation of matrices **A**, **B**, and **C**.

The principle of the ALS algorithm is to convert a non-linear optimization problem into three independent linear

least squares (LS) problems. In the first steps, one of the three matrices, say, **A** is updated while the two others (**B** and **C**) are fixed to their values obtained in previous estimation steps. The estimate of **A** is given by:

$$\hat{\mathbf{A}} = \mathbf{T}^{I,KJ} \left( (\mathbf{C} \odot \mathbf{B})^\dagger \right)^T. \quad (15)$$

where  $\mathbf{T}^{I,KJ}$  is the unfolding matrix of size  $I \times JK$  defined in Section 2.2, and  $()^\dagger$  is the Moore-Penrose pseudo inverse. By symmetry, the expressions are similar for  $\hat{\mathbf{B}}$  and  $\hat{\mathbf{C}}$ .

### 4.2 Proposed algorithms

Our optimization problem consists in minimizing the squared error  $\Upsilon$  under a collection of  $3R$  constraints, namely:

$$\min_{\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{A}} \left\| \mathcal{T} - \sum_{r=1}^R \lambda_r \mathbf{a}_r \otimes \mathbf{b}_r \otimes \mathbf{c}_r \right\|_F^2, \quad (16)$$

$$\|\mathbf{a}_r\| = \|\mathbf{b}_r\| = \|\mathbf{c}_r\| = 1, 1 \leq r \leq R$$

Therefore, we need to find three matrices **A**, **B**, and **C** of unit norm columns which minimize (16). Stack these three matrices in a  $I+J+K$  by  $R$  matrix denoted by **X**. The objective can now be also written  $\Upsilon(\mathbf{X}, \mathbf{A})$ , for the sake of convenience.

The computation of loading matrices is performed by minimizing the quadratic cost function (10). One generates a series of iterates  $\mathbf{X}(k) = [\mathbf{A}(k)^T, \mathbf{B}(k)^T, \mathbf{C}(k)^T]^T$ ,  $k \in \mathbb{N}$ , with initial value  $\mathbf{X}(0)$  arbitrarily chosen. Generally, the algorithm consists of choosing at the  $k$ th iteration a point  $\mathbf{X}(k+1)$  in a direction lying in the half space defined by the gradient of objective function  $\Upsilon$ , defined by matrix **D**( $k$ ), which verifies [22]:

$$\text{vec}\{\nabla \Upsilon(\mathbf{X}(k))\}^T \text{vec}\{\mathbf{D}(k)\} < 0. \quad (17)$$

One possibility is to choose the direction opposite to the gradient:

$$\mathbf{D}(k) = -\nabla \Upsilon(\mathbf{A}(k), \mathbf{B}(k), \mathbf{C}(k)). \quad (18)$$

The gradient components  $\nabla \Upsilon_A$  (size  $I \times R$ ),  $\nabla \Upsilon_B$  ( $J \times R$ ), and  $\nabla \Upsilon_C$  ( $K \times R$ ) can be stacked into a single matrix of size  $I+J+K$  by  $R$ :

$$\mathbf{D} = \begin{pmatrix} -\nabla \Upsilon_A \\ -\nabla \Upsilon_B \\ -\nabla \Upsilon_C \end{pmatrix}. \quad (19)$$

Since objective function (10) is real but its arguments are complex, its gradient can be computed in the sense of

[23,24]. Using the quadratic form proposed in [23], the objective function can be expanded into:

$$\Upsilon(\mathbf{A}, \mathbf{B}, \mathbf{C}; \mathbf{\Lambda}) = \|\mathcal{T}\|^2 + \sum_i \sum_{p=1}^R \sum_{q=1}^R A_{ip} M_{pq} A_{iq}^* - \sum_i \left( \sum_{q=1}^R N_{iq} A_{iq}^* + \sum_{p=1}^R A_{ip} N_{ip}^* \right). \quad (20)$$

where  $M_{pq} = \sum_{jk} \lambda_p \lambda_q^* B_{jp} B_{jq}^* C_{kp} C_{kq}^*$  and  $N_{ip} = \sum_{jk} T_{ijk} \lambda_p^* B_{jp}^* C_{kp}^*$ . Thus, the gradient of  $\Upsilon$  with respect to  $\mathbf{A}$  is:

$$\frac{\partial \Upsilon}{\partial \mathbf{A}} = 2\mathbf{A}\mathbf{M} - 2\mathbf{N}. \quad (21)$$

where  $\mathbf{M}$  of size  $R \times R$  and  $\mathbf{N}$  of size  $I \times R$ . The gradient of  $\Upsilon$  with respect to  $\mathbf{B}$  and  $\mathbf{C}$  is similar, taking into account the fact that matrices  $\mathbf{M}$  and  $\mathbf{N}$  need to be defined accordingly (for the gradient of  $\Upsilon$  with respect to  $\mathbf{B}$  and  $\mathbf{C}$ , the dimension of matrix  $\mathbf{N}$  is  $J \times R$  and  $K \times R$ , respectively, while the dimension of  $\mathbf{M}$  is always  $R \times R$ ).

The difficulty that arises in constrained optimization is to make sure that the move remains within the feasible set,  $\mathcal{C}$ , defined by the constraints. In the following subsections, we propose two versions of our algorithm, with two different ways of calculating scale factor  $\mathbf{\Lambda}$ .

Descent algorithms are also determined by the steps that will be executed in the chosen direction. There are various methods for the step selection, and the most widely used are *Backtracking* and *Armijo* [22]. To compute the stepsize  $\ell(k)$  in Algorithm 1 and Algorithm 2, we use a popular inexact line search method, very simple and quite effective, which is the backtracking line search. It depends on two fixed constants  $\alpha, \beta$  with  $0 < \alpha < 0.5$  and  $0 < \beta < 1$ .

#### Backtracking algorithm

1. Given a descent direction  $\mathbf{D}$  for  $\Upsilon$ ,  $\alpha \in (0, 0.5)$ ,  $\beta \in (0, 1)$ .
2. Initialization:  $\ell = 1$ .
3. while  $\Upsilon(\mathbf{X} + \ell\mathbf{D}; \mathbf{\Lambda}) > \Upsilon(\mathbf{X}; \mathbf{\Lambda}) + \alpha\ell\nabla\Upsilon^T\mathbf{D}$
4.  $\ell = \beta\ell$ .

#### 4.2.1 Algorithm 1

In the recent work on CP tensor decomposition, optimization of the objective function (10) was made without explicitly considering the factor  $\mathbf{\Lambda}$ . More precisely in most contributions, the scaling factor is integrated into loading matrices, so that  $\mathbf{\Lambda}$  may be set to the identity. The first solution we propose is to use a projected gradient

algorithm while calculating  $\mathbf{\Lambda}$  as the product of normalizing factors of matrices  $\mathbf{A}$ ,  $\mathbf{B}$ , and  $\mathbf{C}$ :

$$\mathbf{\Lambda}(k) = \mathbf{\Lambda}(k-1) \square \mathbf{\Lambda}_A \square \mathbf{\Lambda}_B \square \mathbf{\Lambda}_C. \quad (22)$$

where  $\square$  is the Hadamard product,  $\mathbf{\Lambda}_A = \text{Diag}\{\|\mathbf{a}_1\|, \dots, \|\mathbf{a}_R\|\}$ , and similar definitions for  $\mathbf{\Lambda}_B$  and  $\mathbf{\Lambda}_C$ . This approach, which we call 'Algorithm 1' can be described by the pseudo-code below:

1. Initialize  $\{\mathbf{A}(0); \mathbf{B}(0); \mathbf{C}(0)\}$  to full-rank matrices with unit norm columns, and set  $\mathbf{\Lambda}(0) = \mathbf{I}$
2. For  $k \geq 1$  and subject to a stopping criterion, do:

- (a) Compute the descent direction as the gradient w.r.t.  $\mathbf{X}$ :

$$\mathbf{D}(k) = -\nabla\Upsilon(\mathbf{X}(k-1); \mathbf{\Lambda}(k-1)),$$

- (b) Compute a stepsize  $\ell(k)$  using the backtracking method [22] such as:

$$\begin{aligned} &\Upsilon(\mathbf{X}(k-1) + \ell(k)\mathbf{D}(k); \mathbf{\Lambda}(k-1)) \\ &< \Upsilon(\mathbf{X}(k-1); \mathbf{\Lambda}(k-1)). \end{aligned}$$

- (c) Update  $\mathbf{X}(k) = \mathbf{X}(k-1) + \ell(k)\mathbf{D}(k)$
- (d) Extract the three blocks of  $\mathbf{X}(k)$ :  $\mathbf{A}(k)$ ,  $\mathbf{B}(k)$  and  $\mathbf{C}(k)$ , and store the norm of their columns into  $\mathbf{\Lambda}_A$ ,  $\mathbf{\Lambda}_B$ ,  $\mathbf{\Lambda}_C$
- (e) Normalize them to unit-norm columns as:

$$\begin{aligned} \mathbf{A}(k) &:= \mathbf{A}(k)\mathbf{\Lambda}_A^{-1}, \mathbf{B}(k) := \mathbf{B}(k)\mathbf{\Lambda}_B^{-1}, \\ &\text{and } \mathbf{C}(k) := \mathbf{C}(k)\mathbf{\Lambda}_C^{-1} \end{aligned}$$

- (f) update  $\mathbf{\Lambda}(k) = \mathbf{\Lambda}(k-1) \square \mathbf{\Lambda}_A \square \mathbf{\Lambda}_B \square \mathbf{\Lambda}_C$

#### 4.2.2 Algorithm 2

The other approach is to consider  $\mathbf{\Lambda}$  as an additional variable. By canceling the gradient of  $\Upsilon(\mathbf{X}, \mathbf{\Lambda})$  with respect to  $\mathbf{\Lambda}$ , one obtains Equation 11, which can be solved for  $\mathbf{\Lambda}$  when  $\mathbf{X}$  is fixed. This gives the algorithms below.

#### Algorithm 2.

1. Initialize  $(\mathbf{A}(0), \mathbf{B}(0), \mathbf{C}(0))$  to full-rank matrices with unit-norm columns.
2. Compute  $\mathbf{G}(0)$  and  $\mathbf{f}(0)$ , and solve  $\mathbf{G}(0)\mathbf{\lambda} = \mathbf{f}(0)$  for  $\mathbf{\lambda}$ , as defined in Section 3.2 by (11). Set  $\mathbf{\Lambda}(0) = \text{Diag}\{\mathbf{\lambda}\}$ .
3. For  $k \geq 1$  and subject to a stopping criterion, do
  - (a) Compute the descent direction as the gradient w.r.t.  $\mathbf{X}$ :

$$\mathbf{D}(k) = -\nabla\Upsilon(\mathbf{X}(k-1); \mathbf{\Lambda}(k-1))$$

- (b) Compute a stepsize  $\ell(k)$  using the backtracking method [22] such as:

$$\begin{aligned} &\Upsilon(\mathbf{X}(k-1) + \ell(k)\mathbf{D}(k); \mathbf{\Lambda}(k-1)) \\ &< \Upsilon(\mathbf{X}(k-1); \mathbf{\Lambda}(k-1)). \end{aligned}$$

- (c) Update  $\mathbf{X}(k) = \mathbf{X}(k-1) + \ell(k) \mathbf{D}(k)$
- (d) Extract the three blocks of  $\mathbf{X}(k)$ :  $\mathbf{A}(k)$ ,  $\mathbf{B}(k)$  and  $\mathbf{C}(k)$
- (e) Normalize the columns of  $\mathbf{A}(k)$ ,  $\mathbf{B}(k)$  and  $\mathbf{C}(k)$
- (f) Compute  $\mathbf{G}(k)$  and  $\mathbf{f}(k)$ , and solve  $\mathbf{G}(k) \boldsymbol{\lambda} = \mathbf{f}(k)$  for  $\boldsymbol{\lambda}$ , according to (11). Set  $\boldsymbol{\Lambda}(k) = \text{Diag}\{\boldsymbol{\lambda}\}$ .

**Stopping criterion.** The convergence is usually considered to be obtained at the  $k$ th iteration when the error between tensor  $\mathcal{T}$ , and the tensor reconstructed from the estimated loading matrices, does not significantly change between iterations  $k$  and  $k+1$ .

However, in the complex case, the phase of the entries of loading matrices found at the end of the algorithm – as defined by the stopping criterion above – is different from the original. To remedy this problem, we propose a new performance criterion in order to minimize the distance between the original and the estimated matrices. Although this criterion is not usable when actual loading matrices are unknown, it still permits to assess the performances effectively attained.

## 5 Performance criterion

As highlighted in Section 2, there is always an indeterminacy in the representation of decomposable tensors, characterized in the CP decomposition by  $3R$  complex numbers of unit modulus. In order to better understand this problem, let  $\mathbf{a}$ ,  $\mathbf{b}$ , and  $\mathbf{c}$  denote the  $r$ th column of matrices  $\mathbf{A}$ ,  $\mathbf{B}$ , and  $\mathbf{C}$ , respectively, with  $1 \leq r \leq R$ . Furthermore,  $\hat{\mathbf{a}}$ ,  $\hat{\mathbf{b}}$ , and  $\hat{\mathbf{c}}$  denote one column of the estimated matrices entering in the CP decomposition. We seek to minimize a distance:

$$\delta(\mathbf{x}; \hat{\mathbf{x}}) = \min_{\varphi, \psi, \chi} \left\{ \|\mathbf{a} - e^{j\varphi} \hat{\mathbf{a}}\|^2 + \|\mathbf{b} - e^{j\psi} \hat{\mathbf{b}}\|^2 + \|\mathbf{c} - e^{j\chi} \hat{\mathbf{c}}\|^2 \right\}. \quad (23)$$

In the literature, only approximate performance indices have been used so far by neglecting relation between the angles  $\varphi$ ,  $\psi$  and  $\chi$  given by:

$$\exp(j(\varphi + \psi + \chi)) = 1.$$

Our contribution herein consists in finding the exact minimum distance (23) under this angular constraint, by calculating the  $3R$  optimal phases affecting the columns of the estimated loading matrices.

The derivative of Equation 23 with respect to  $\varphi$  and  $\psi$  yields a system of two equations. Finding  $3R$  phases means to solve, if the 2 sets of 3 columns are fixed:

$$\begin{cases} \rho_a \sin x + \rho_c \sin(\varphi + \psi + \gamma) &= 0, \\ \rho_b \sin y + \rho_c \sin(\varphi + \psi + \gamma) &= 0. \end{cases} \quad (24)$$

where  $x = \varphi - \alpha$  and  $y = \psi - \beta$ . After some trigonometric manipulations described in Appendix 2, the solutions can be obtained by rooting a polynomial of degree 6 in a single variable  $\varphi$ . By replacing the admissible values of  $\varphi$  into system (24), the corresponding values of  $\psi$  are obtained. The calculation of the minimum distance (23) is done for all possible permutations. We end up with the following performance criterion:

$$\mathcal{E}(\mathcal{T}; \mathbf{A}, \mathbf{B}, \mathbf{C}, \boldsymbol{\Lambda}) = \min_{\pi \in \Pi} \sum_{r=1}^R \delta(\mathbf{x}_r; \hat{\mathbf{x}}_{\pi(r)}). \quad (25)$$

where  $\Pi$  is the set of permutations of  $\{1, 2, \dots, R\}$ . When the permutation acts in too large dimension, greedy versions are possible to limit the exhaustive search in the permutation set.

Denote  $\pi(i)$  as the permutations of  $\Pi$ ,  $1 \leq i \leq R!$ . The overall algorithm to compute the performance criterion is summarized as follows:

1. For  $1 \leq i \leq R!$  do
2. Calculate the  $3R$  optimal phases affecting the columns of the estimated loading matrices:
  - (a) Permute the columns of the three estimated matrices according to the permutation  $\pi(i)$ :  $\hat{\mathbf{A}}_{\pi(i)}$ ,  $\hat{\mathbf{B}}_{\pi(i)}$ , and  $\hat{\mathbf{C}}_{\pi(i)}$ ;
  - (b) For each  $r$ th columns of loading matrices and estimated matrices, do:
    - Set  $x = \varphi - \alpha$  and  $y = \psi - \beta$  and solve the polynomial of degree 6 in a single variable  $\varphi$ :
$$c_0 + c_1 \cos(2x) + c_2 \cos^2(2x) + c_3 \cos^3(2x) + c_4 \cos^4(2x) + c_5 \cos^5(2x) + c_6 \cos^6(2x) = 0.$$
    - Replace  $x$  in  $\sin y = \frac{\rho_a}{\rho_b} \sin x$ , obtain  $y$  and consequently  $\psi$ ;
    - Calculate  $\chi$  in:  $\exp(j(\varphi + \psi + \chi)) = 1$ ;
    - Calculate the minimum distance  $\delta$ :
$$\delta(\mathbf{x}; \hat{\mathbf{x}}) = \min_{\varphi, \psi, \chi} \left\{ \|\mathbf{a} - e^{j\varphi} \hat{\mathbf{a}}\|^2 + \|\mathbf{b} - e^{j\psi} \hat{\mathbf{b}}\|^2 + \|\mathbf{c} - e^{j\chi} \hat{\mathbf{c}}\|^2 \right\}.$$
  - Save the results: distance( $i$ ) =  $\delta$ , phase $_{\varphi}$ ( $i$ ) =  $\varphi$ , phase $_{\psi}$ ( $i$ ) =  $\psi$  and phase $_{\chi}$ ( $i$ ) =  $\chi$ ;
3. End do.
4. Choose the  $3R$  angles which return the smaller distance  $\delta$ .

## 6 Simulation results

To evaluate the efficiency and behavior of the proposed algorithms with new performance criterion, two experiments are made: the first one for random loading matrices

and the second one for DS-CDMA system. In all experiments, the results are obtained from 100 Monte Carlo runs. At each iteration and for every SNR value, a new noise realization is drawn. The stopping criterion chosen for all experiments is  $\Upsilon^{(n)} < \epsilon$  and  $|\Upsilon^{(n)} - \Upsilon^{(n-1)}| < \epsilon$ , where  $\epsilon$  is a threshold by which the global minimum is considered to be reached, and  $n$  is the current iteration. In the following simulations, we take:  $\epsilon = 10^{-6}$ .

### 6.1 Example 1: random loading matrices

In order to illustrate the behavior and performances of the proposed algorithms, we first report simulation results run on random tensors. In a first stage, we compare 'Algorithm 2', 'Algorithm 1', and the gradient descent algorithm named herein 'Algorithm 0'. We will analyze the errors of matrix estimation obtained through the performance criterion proposed in Section 5. Two scenarios are analyzed using random tensors: (i) one tensor of rank 2 with dimensions  $3 \times 3 \times 3$  and (ii) another tensor of rank 5 with dimensions  $8 \times 8 \times 8$ . Loading matrices are initialized with random-valued columns and the two tensors are corrupted by an additive Gaussian noise.

Figures 1 and 2 depict matrix estimation errors implied in (25) as a function of SNR. As expected, it can be seen that the results using Algorithm 0 show a poor performance when compared to the results obtained with Algorithms 1 and 2 (curves with diamonds and circles). This supports the idea that our algorithms isolating the scaling matrix are attractive. Furthermore, we check that when the phase constraints are neglected in the calculation of the performance measure, the results are significantly more optimistic, particularly at high SNR (curves of the same color), which supports the interest of using our performance index defined in (5).

In order to show the significant difference between Algorithms 1 and 2, we will examine the convergence speed of the tensor reconstruction according to the number of iterations, since the final error is about the same (cf. Figures 1 and 2). We consider the case of a  $3 \times 3 \times 3$  tensor of rank 2. The latter was constructed from three  $3 \times 2$  Gaussian matrices drawn randomly. The results are presented in Figure 3, and show that in all our experiments, Algorithm 2 converged faster in terms of number of iterations, while Algorithm 1 and Algorithm 0 require more iterations to reach convergence.

### 6.2 Example 2: application To DS-CDMA system

In this example, we place ourselves in a blind context. We assume that the receiver has no knowledge neither on the spreading codes nor on symbol sequences. Classically, telecommunications blind techniques are based on some *a priori* knowledge, such as temporal properties of transmitted signals or the spatial properties of the receiver [25-27].

Recently, algebraic tensor methods have received considerable attention in signal processing [2]. It also turns out that multilinear algebra tools are often more powerful than their matrix equivalent. Sidiropoulos et al. are the first to adopt tensor approaches in the telecommunications field in 2000 [12]. They observed that the samples of a CDMA signal received by an array of antennas can be stored in a cube, each dimension corresponding to a diversity (coding diversity, temporal diversity, and spatial diversity). Thus, they showed that the deterministic blind separation problem of CDMA signals can be solved by the CP decomposition [28].

In this example, we propose to apply the CP decomposition algorithms as detailed in Section 4 with the new performance criterion on the DS-CDMA technique. A comparison with the ALS algorithm is then made.

#### 6.2.1 Tensor modeling

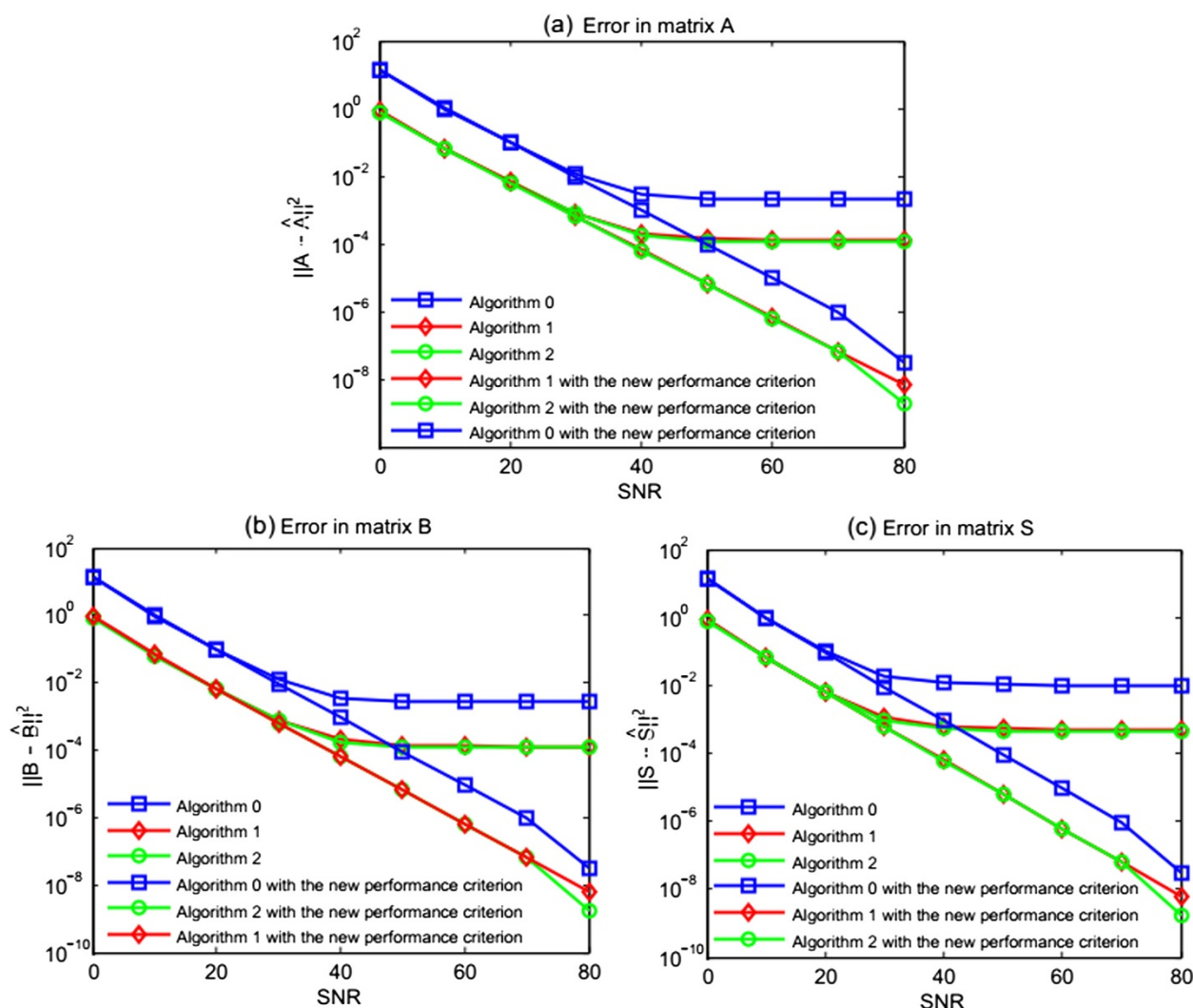
We consider  $R$  users with one transmitting antenna, transmitting simultaneously their signals to an array of  $K$  receiving antennas. In other words, we consider a communication system of type 'multiuser SIMO'.

For example, assume the user  $r$  transmits the symbols  $\mathbf{s}_r$  of size  $J$ . These symbols are spreaded by  $\mathbf{c}_r$  CDMA code of length  $I$ , uniquely allocated to user  $r$  and assumed to be unknown at the receiver. These codes are binary sequences taking values from  $\{-1, 1\}$  and could be non-orthogonal. The spreading sequence propagates along a single path in a memoryless channel and is received by the antenna array under angle of arrival  $\theta_r$ . Each of the  $K$  antennas receives a signal  $\mathbf{Y}_k(t)$  of size  $J \times I$ . Our approach for detection and separation of the received signals is to exploit the multilinear algebraic structure of these signals using the new performance criterion. We observe these signals during a time span of length  $J/T_s$ , where  $T_s$  is the symbol period. The  $\mathbf{Y}_k(t)$  signals are sampled at the chip period  $T_c = T_s/I$ , where  $I$  is the spreading factor. Therefore, each antenna provides a set of  $IJ$  samples that can be ordered in a tensor of order 3, denoted  $\mathcal{Y} \in \mathbb{C}^{I \times J \times K}$ . The  $Y_{ijk}$  component of this tensor that corresponds to the sample of the overall signal received by the  $k$ th antenna at time  $i$  of the  $j$ th symbol period can be written as follows:

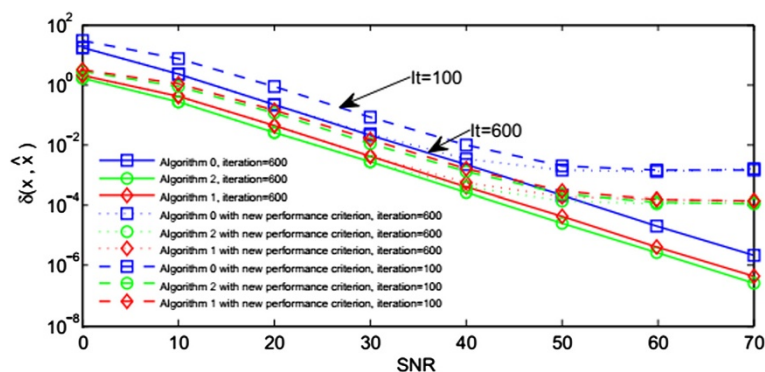
$$Y_{ijk} = \sum_{r=1}^R A_{kr} S_{jr} C_{ir} \quad i \in [1, I] \quad j \in [1, J] \quad k \in [1, K]. \quad (26)$$

where the complex scalar  $A_{kr} = \beta_r \mathbf{a}_k(\theta_r)$ , with  $\beta_r$  is the fading coefficient of the  $r$ th user and  $\mathbf{a}_k(\theta_r)$  the response of the antenna  $k$  at the angle of arrival  $\theta_r$ .

Separating the received signals is then equivalent to decompose the tensor  $\mathcal{Y}$  into a sum of  $R$  contributions, where  $R$  represents the number of active users in the

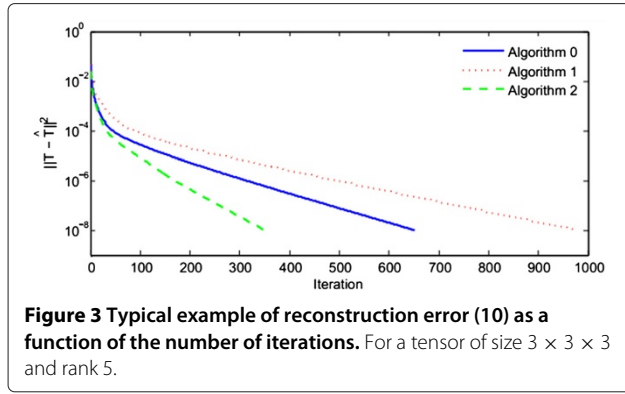


**Figure 1** Matrix estimation errors, with a random tensor of size  $8 \times 8 \times 8$  and rank 5. (a) Matrix A. (b) Matrix B. (c) Matrix S.



**Figure 2** Sum of matrix estimation errors, with a random tensor of size  $3 \times 3 \times 3$  and rank 5. Note the asymptote depending on the maximum number of iterations executed.





system. To calculate the CP decomposition of  $\mathcal{Y}$ , we resort to Algorithm 2 presented in Section 4 with performance index (25). The detection and separation of the matrix  $\mathbf{S}$  of transmitted symbols will be made, using the following objective function:

$$\Upsilon(\mathbf{A}, \mathbf{C}, \mathbf{S}, \mathbf{\Lambda}) = \left\| \mathcal{Y} - \sum_{r=1}^R \lambda_r \mathbf{a}_r \otimes \mathbf{c}_r \otimes \mathbf{s}_r \right\|_F^2. \quad (27)$$

where  $\mathbf{a}_r$ ,  $\mathbf{c}_r$ , and  $\mathbf{s}_r$  are the normalized vectors; the scaling ambiguities on the estimated symbols are eliminated by normalizing each symbol sequence by its norm and calculating the exact scaling factor  $\mathbf{\Lambda}$ . As to correct the phases of the three estimated matrices, we will use the exact performance index (28) seen in Section 5.

$$\mathcal{E}(\mathcal{T}; \mathbf{A}, \mathbf{C}, \mathbf{S}, \mathbf{\Lambda}) = \min_{\pi \in \Pi} \sum_{r=1}^R \left( \min_{\varphi, \psi, \chi} \left\{ \left\| \mathbf{a}_r - e^{j\varphi} \hat{\mathbf{a}}_{\pi(r)} \right\|^2 + \left\| \mathbf{c}_r - e^{j\psi} \hat{\mathbf{c}}_{\pi(r)} \right\|^2 + \left\| \mathbf{s}_r - e^{j\chi} \hat{\mathbf{s}}_{\pi(r)} \right\|^2 \right\} \right). \quad (28)$$

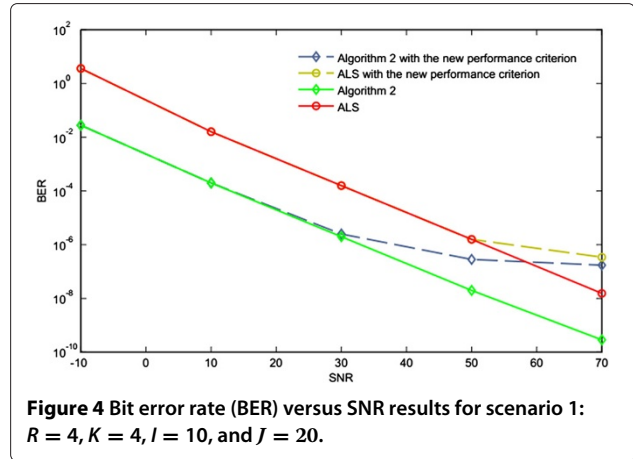
### 6.2.2 Simulation

In this experiment, we present the performance of the receiver algorithms (Algorithm 1 and Algorithm 2 with new performance criterion) which estimate blindly the symbol  $\mathbf{S}$ .

We consider  $R = 4$  users communicate simultaneously in the same bandwidth. Each user transmits a sequence  $J = 20$  of consecutive symbols and is uniquely assigned

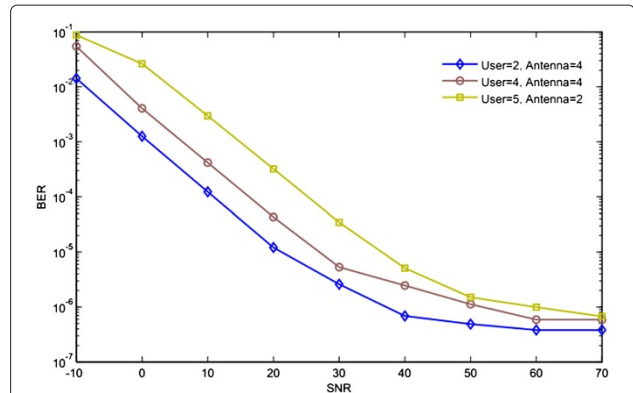
**Table 1** Angles of arrival for four users

	Angles of arrival
Figure 4	(-60, -30, 0, 20)



a spreading sequence of length  $I = 10$ . The user symbols are generated from an i.i.d distribution and are modulated using a pseudo-random quaternary phase shift keying (QPSK) sequence. The signal is transmitted to a receiver of  $K = 4$  antennas. The angles of arrival are described in Table 1.

In Figure 4, we will illustrate the ability of the blind tensorial receiver Algorithm 2 using the new performance criterion and the ALS receiver for noisy data extraction. Using Monte Carlo simulations, we will represent the evolution of bit error rate (BER) according to the signal-to-noise ratio (SNR). These results shown in Figure 4 indicate that the performance of the proposed receiver Algorithm 2 is better since it converges faster than the ALS algorithm. That is tied to the normalization of the factor matrices and the exact calculation of the scaling factor. Moreover, we can see that BER of Algorithm 2 without the use of our criterion performance is very optimistic, which proves the interest of our study.



**Table 2 Angles of arrival for three scenarios**

	Angles of arrival
Curve 1	(-10, 20)
Curve 2	(-60, -40, 0, 20)
Curve 3	(-60, -50, 10, 40, 80)

The impact of factor  $\frac{K}{R}$ , which represents the number of receiving antennas per user is illustrated in Figure 5. In this figure, curve 1 represents the case where the number of antenna is  $K = 4$  and the number of users is  $R = 2$ ,  $K = 4$ , and  $R = 4$  for the second curve, while for the third curve  $K = 2$  and  $R = 5$ . The angles of arrival are described in Table 2. As a result, the overall system performance is enhanced when increasing the factor  $\frac{K}{R}$ , indicating the importance of spatial diversity.

## 7 Conclusions

In this paper, we have shown in Section 3.2 that, in CP tensor decompositions, the scale matrix  $\mathbf{\Lambda}$  takes as optimal value a Gram matrix controlling the conditioning of the problem. This shows that bounding coherences would allow to ensure a minimal conditioning. We have described several algorithms able to compute the minimal polyadic decomposition of three-way arrays. The two proposed algorithms Algorithm 1 and Algorithm 2 have been described and tested, which involve a separate explicit calculation of the scale matrix  $\mathbf{\Lambda}$ . Contrary to the performance measures used in the literature, which are optimistic by construction, the performance index calculated herein is more realistic by taking into account the angular constraint. An application of the CP decomposition with exact performance criterion to DS-CDMA system has been presented. Finally, computer simulations have been performed in the context of SIMO-CDMA system, in order to demonstrate both the good performances of the proposed algorithms, compared to ALS one and their usefulness in CDMA system. The judgment of our algorithms do not solely rely on the reconstruction error and the convergence speed, but it also takes into account the error in the loading matrices obtained and the BER in the case of the CDMA application.

## Appendices

### Appendix 1

#### Detailed $\mathbf{\Lambda}$ optimal calculation

We intend to calculate the optimal value of  $\mathbf{\Lambda}$  which minimize the following expression:

$$\Upsilon(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{\Lambda}) = \|\mathcal{T} - (\mathbf{A}, \mathbf{B}, \mathbf{C}) \cdot \mathbf{\Lambda}\|_F^2. \quad (29)$$

By developing, it leads to:

$$\begin{aligned} \Upsilon(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{\Lambda}) &= \|\mathcal{T}\|^2 - \sum_{ijk} \sum_p T_{ijk}^* \lambda_p A_{ip} B_{jp} C_{kp} \\ &\quad - \sum_{ijk} \sum_q T_{ijk} \lambda_q^* A_{iq}^* B_{jq}^* C_{kq}^* \\ &\quad + \sum_{pq} \sum_{ijk} \lambda_p A_{ip} B_{jp} C_{kp} \lambda_q^* A_{iq}^* B_{jq}^* C_{kq}^* \\ &= \|\mathcal{T}\|^2 - \sum_p \lambda_p f_p^* - \sum_q \lambda_q^* f_q + \sum_{pq} \lambda_p \lambda_q^* G_{pq}. \end{aligned} \quad (30)$$

where  $G_{pq} = \sum_{ijk} A_{ip} B_{jp} C_{kp} A_{iq}^* B_{jq}^* C_{kq}^*$  and  $f_q = \sum_{ijk} T_{ijk} A_{iq}^* B_{jq}^* C_{kq}^*$ . By canceling the derivative of  $\Upsilon$  with respect to  $\lambda$ , we find the following linear system:

$$\mathbf{G}\mathbf{\Lambda} = \mathbf{f}. \quad (31)$$

## Appendix 2

### Performance criterion details

In this appendix, we explain in more details how to obtain performance index  $\delta$ , in particular how phases  $(\varphi, \psi, \chi)$  are calculated. Setting  $\chi = -\varphi - \psi [2\pi]$ , equation (23) can be rewritten as:

$$\begin{aligned} \delta &= \|\mathbf{a}\|^2 + \|\hat{\mathbf{a}}\|^2 + \|\mathbf{b}\|^2 + \|\hat{\mathbf{b}}\|^2 + \|\mathbf{s}\|^2 + \|\hat{\mathbf{s}}\|^2 \\ &\quad - 2\rho_{\mathbf{a}} \cos(\varphi - \alpha) - 2\rho_{\mathbf{b}} \cos(\psi - \beta) \\ &\quad - 2\rho_{\mathbf{s}} \cos(\varphi + \psi + \gamma). \end{aligned}$$

where  $\mathbf{a}^H \hat{\mathbf{a}} \stackrel{\text{def}}{=} \rho_{\mathbf{a}} e^{j\alpha}$ ,  $\mathbf{b}^H \hat{\mathbf{b}} \stackrel{\text{def}}{=} \rho_{\mathbf{b}} e^{j\beta}$  and  $\mathbf{s}^H \hat{\mathbf{s}} \stackrel{\text{def}}{=} \rho_{\mathbf{s}} e^{j\gamma}$ . Stationary points are given by the solutions of the trigonometric system in *e.g.* variables  $x = \varphi - \alpha$  and  $y = \psi - \beta$  as unknowns:

$$\begin{aligned} \rho_{\mathbf{a}} \sin x + \rho_{\mathbf{s}} \sin(\varphi + \psi + \gamma) &= 0, \\ \rho_{\mathbf{b}} \sin y + \rho_{\mathbf{s}} \sin(\varphi + \psi + \gamma) &= 0. \end{aligned}$$

The first simplification is achieved by noting that

$$\rho_{\mathbf{s}} \sin(\varphi + \psi + \gamma) = -\rho_{\mathbf{a}} \sin x = -\rho_{\mathbf{b}} \sin y.$$

implies

$$\sin y = \frac{\rho_{\mathbf{a}}}{\rho_{\mathbf{b}}} \sin x.$$

Now, using trigonometric identities, we can rewrite the first equation of the trigonometric system

$$\rho_{\mathbf{s}} \sin(\varphi + \psi + \gamma) = \rho_{\mathbf{s}} \sin(x + y + \alpha + \beta + \gamma) = -\rho_{\mathbf{a}} \sin x.$$

as

$$\begin{aligned} \rho_{\mathbf{a}} \sin x &= -\rho_{\mathbf{s}} [\sin x \cos y \cos(\alpha + \beta + \gamma) \\ &\quad + \sin y \cos x \cos(\alpha + \beta + \gamma) \\ &\quad + \cos x \cos y \sin(\alpha + \beta + \gamma) \\ &\quad - \sin x \sin y \sin(\alpha + \beta + \gamma)]. \end{aligned}$$

Letting  $\cos y = \sqrt{1 - \sin^2 y}$  and  $\sin y = \frac{\rho_a}{\rho_b} \sin x$ , we obtain and

$$\begin{aligned} \rho_a \sin x &= -\frac{\rho_s \rho_a}{\rho_b} \sin x \cos x \cos(\alpha + \beta + \gamma) \\ &+ \frac{\rho_s \rho_a}{\rho_b} \sin^2 x \sin(\alpha + \beta + \gamma) \\ &+ [\rho_s \sin x \sin(\alpha + \beta + \gamma) \\ &- \rho_s \cos x \sin(\alpha + \beta + \gamma)] \sqrt{1 - \frac{\rho_a}{\rho_b} \sin^2 x}. \end{aligned}$$

The goal of the next step is to eliminate the square root and to rewrite the equation in term of variables  $\sin x$  or  $\cos x$ . So, let us squaring both side of this equation and using trigonometric identities such as  $\cos^2 x = \frac{1+\cos(2x)}{2}$ ,  $\sin^2 x = \frac{1-\cos(2x)}{2}$ ,  $\cos x \sin x = \frac{\sin(2x)}{2}$ , and  $\cos^2(2x) + \sin^2(2x) = 1$ . Thus, after simplification we obtain

$$\begin{aligned} \frac{\rho_b^2}{2} + \frac{1}{2} \left( \frac{\rho_s \rho_a}{\rho_b} \right)^2 - \frac{\rho_s^2}{2} + \left[ \frac{\rho_b^2}{2} + \frac{1}{2} \left( \frac{\rho_s \rho_a}{\rho_b} \right)^2 \right. \\ \left. + \frac{\rho_s^2}{2} \cos^2(\alpha + \beta + \gamma) - \frac{\rho_s^2}{2} \sin^2(\alpha + \beta + \gamma) \right] \cos^2(2x) \\ + [2\rho_s \cos(\alpha + \beta + \gamma) \sin(\alpha + \beta + \gamma)] \sqrt{1 - \cos^2(2x)} \\ + \left[ 2 \left( \frac{\rho_s \rho_a^2}{\rho_b} \right) \cos(\alpha + \beta + \gamma) \sqrt{1 - \cos^2(2x)} \right. \\ \left. - \left( \frac{\rho_s \rho_a^2}{\rho_b} \right) \sin(\alpha + \beta + \gamma) (1 - \cos(2x)) \right] \sqrt{\frac{1 - \cos(2x)}{2}} \\ = 0. \end{aligned}$$

In the same way as above, we squared twice both sides of the resulting equation to eliminate the squares roots. Finally, we get an equation of degree six of the form

$$\begin{aligned} c_0 + c_1 \cos(2x) + c_2 \cos^2(2x) + c_3 \cos^3(2x) \\ + c_4 \cos^4(2x) + c_5 \cos^5(2x) + c_6 \cos^6(2x) = 0. \end{aligned}$$

with

$$\begin{cases} c_0 &= c_1'^2 - c_5'^2, \\ c_1 &= 2c_1'c_4' - 2c_5'c_7', \\ c_2 &= 2c_1'c_3' + c_4'^2 - 2c_5'c_6' - c_7'^2 + c_5'^2, \\ c_3 &= 2c_1'c_2' + 2c_3'c_4' - 2c_6'c_7' + 2c_5'c_7', \\ c_4 &= c_3'^2 + 2c_2'c_4' - c_6'^2 + 2c_5'c_6' + c_7'^2, \\ c_5 &= 2c_2'c_3' + 2c_6'c_7', \\ c_6 &= c_2'^2 + c_6'^2. \end{cases}$$

$$\begin{cases} c_1' &= c_1''^2 + c_3''^2 - \frac{1}{2}c_4''^2 - \frac{1}{2}c_5''^2, \\ c_2' &= -\frac{1}{2}c_4''^2 + \frac{1}{2}c_5''^2, \\ c_3' &= c_2''^2 - c_3''^2 + \frac{1}{2}c_4''^2 - \frac{3}{2}c_5''^2, \\ c_4' &= 2c_1''c_2' + \frac{1}{2}c_4''^2 + \frac{4}{2}c_5''^2, \\ c_5' &= -2c_1''c_3' + c_4'c_5', \\ c_6' &= c_4'c_5'', \\ c_7' &= -2c_2''c_3' - 2c_4'c_5''. \end{cases}$$

and

$$\begin{cases} c_1'' &= \frac{1}{2}\rho_a^2 + \frac{1}{2} \left( \frac{\rho_s \rho_a}{\rho_b} \right)^2 - \frac{1}{2}\rho_s^2, \\ c_2'' &= -\frac{1}{2}\rho_a^2 - \frac{1}{2} \left( \frac{\rho_s \rho_a}{\rho_b} \right)^2 + \rho_s^2 \cos^2(\alpha + \beta + \gamma) - \frac{1}{2}, \\ c_3'' &= -2\rho_s^2 \cos(\alpha + \beta + \gamma) \sin(\alpha + \beta + \gamma), \\ c_4'' &= -2\frac{\rho_s \rho_a^2}{\rho_b} \cos(\alpha + \beta + \gamma), \\ c_5'' &= \frac{\rho_s \rho_a^2}{\rho_b} \sin(\alpha + \beta + \gamma). \end{cases}$$

Solving the sixth degree equation yields  $x$ . Replacing  $x$  in  $\sin y = \frac{\rho_a}{\rho_b} \sin x$  yields  $y$ .

#### Competing interests

The authors declare that they have no competing interests.

#### Author details

<sup>1</sup>Mohammed V-Agdal University, LRIT Associated Unit to the CNRST-URAC N29, Avenue des nations unies, Agdal, BP. 554 Rabat-Chellah, Morocco.

<sup>2</sup>GIPSA-Lab, CNRS UMR5216, Grenoble Campus, St Martin d'Heres cedex, BP. 46 Grenoble 38402, France.

Received: 15 January 2014 Accepted: 6 August 2014

Published: 15 August 2014

#### References

1. C Jutten, J Héroult, Blind separation of sources, part I: an adaptive algorithm based on neuromimetic architecture. *Signal Process.* **24**, 1–20 (1991)
2. P Comon, C Jutten, *Handbook of Blind Source Separation, Independent Component Analysis and Applications*. (Academic Press, Oxford UK, Burlington USA, 2010). ISBN: 978-0-12-374726-6
3. RA Harshman, Foundations of the Parafac procedure: models and conditions for an "explanatory" multi-modal factor analysis. UCLA working papers in phonetics, (University Microfilms, Ann Arbor, Michigan, No. 10,085). **16**, 1–84 (1970)
4. P Comon, Tensors: a brief introduction. *IEEE Sig. Proc. Mag.* **31**(3), 44–53 (2014). Special issue on BSS. hal-00923279
5. TG Kolda, BW Bader, Tensor decompositions and applications. *SIAM Rev.* **51**(3), 455–500 (2009)
6. P Comon, Tensor decompositions, state of the art and applications, in *IMA Conf. Mathematics in Signal Processing* (Clarendon press, Oxford, UK, 2000), pp. 1–24
7. ALFD Almeida, G Favier, JCM Mota, The constrained trilinear decomposition with application to MIMO wireless communication systems, in *GRETSI'07. Colloque GRETSI* (Troyes, 2007)
8. P Comon, X Luciani, ALF De Almeida, Tensor decompositions, alternating least squares and other tales. *J. Chemometrics* **23**, 393–405 (2009)
9. L Sorber, M Van Barel, L De Lathauwer, Optimization-based algorithms for tensor decompositions: canonical polyadic decomposition, decomposition in rank-s terms and a new generalization. *SIAM J. Optimization* **23**(2), 695–720 (2013)

10. P Comon, K Minaoui, A Rouijel, D Aboutajdine, Performance index for tensor polyadic decompositions, in *21th EUSIPCO Conference* (IEEE, Marrakech, Morocco, 2013)
11. JMFT Berge, N Sidiropoulos, On uniqueness in candecomp/parafac. *Psychometrika* **67**(3), 399–409 (2002)
12. ND Sidiropoulos, R Bro, GB Giannakis, Parallel factor analysis in sensor array processing. *IEEE Trans. Sig. Proc.* **48**(8), 2377–2388 (2000)
13. JB Kruskal, Three-way arrays: rank and uniqueness of trilinear decompositions. *Linear Algebra Appl.* **18**, 95–138 (1977)
14. JMF ten Berge, ND Sidiropoulos, On uniqueness in candecomp/parafac. *Psychometrika* **67**(3), 399–409 (2002)
15. L-H Lim, P Comon, Blind multilinear identification. *IEEE Trans. Inf. Theory* **60**(2), 1260–1280 (2014)
16. P Comon, L-H Lim, Sparse representations and low-rank tensor approximation. Research Report ISRN I3S//RR-2011-01-FR, I3S, Sophia-Antipolis, France (February 2011.) <http://hal.archives-ouvertes.fr/docs/00/70/34/94/PDF/RR-11-02-P.COMON.pdf>
17. L-H Lim, P Comon, Multirray signal processing: tensor decomposition meets compressed sensing. *Compte-Rendus Mécanique de l'Académie des Sci.* **338**(6), 311–320 (2010)
18. R Gribonval, M Nielsen, Sparse representations in unions of bases. *IEEE Trans. Inf. Theory* **49**(13), 3320–3325 (2003)
19. E Acar, DM Dunlavy, TG Kolda, A scalable optimization approach for fitting canonical tensor decompositions. *J. Chemometrics* **25**, 67–86 (2011)
20. J Carroll, J Chang, Analysis of individual differences in multidimensional scaling via an n-way generalization of “eckart-young” decomposition. *Psychometrika* **35**, 283–319 (1970)
21. A Smilde, R Bro, P Geladi, *Multi-Way Analysis*. (Wiley, Chichester UK, 2004)
22. S Boyd, L Vandenberghe, *Convex Optimization*. (Cambridge University Press, the United States of America, New York, 2004). ISBN: 978-521-83378-3 hardback
23. P Comon, Estimation multivariable complexe. *Traitement du Signal* **3**(2), 97–101 (1986)
24. A Hjørungnes, D Gesbert, Complex valued matrix differentiation: techniques and key results. *IEEE Trans. Signal Process.* **55**(6), 2740–2746 (2007)
25. E Moulines, P Duhamel, JF Cardoso, S Mayrargue, Subspace methods for the blind identification of multichannel fir filters. *IEEE Trans. Signal Process.* **43**, 516–525 (1995)
26. DN Godard, Self-recovering equalization and carrier tracking in two-dimensional data communication systems. *IEEE Trans. Commun.* **28**, 1867–1875 (1980)
27. AJV der Veen, A Paulraj, An analytical constant modulus algorithm. *IEEE Trans. Signal Proc.* **44**, 1136–1155 (1996)
28. ALF de Almeida, G Favier, JCM Mota, Parafac-based unified tensor modeling for wireless communication systems with application to blind multiuser equalization. *Signal Process.* **87**(2), 337–351 (2007)

doi:10.1186/1687-6180-2014-128

**Cite this article as:** Rouijel et al.: CP decomposition approach to blind separation for DS-CDMA system using a new performance index. *EURASIP Journal on Advances in Signal Processing* 2014 **2014**:128.

**Submit your manuscript to a SpringerOpen<sup>®</sup> journal and benefit from:**

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

---

Submit your next manuscript at ► [springeropen.com](http://springeropen.com)